Deep Time Warping for Multiple Time Series Alignment

Alireza Nourbakhsh[1] and Hoda Mohammadzade[2*]

[1]Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran.
Tel: +98 919 572 6063, Email: alireza.nourbakhsh@ee.sharif.edu
[2]Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran.
Tel: +98 912 711 4912, Email: hoda@sharif.edu

Abstract

Time Series Alignment is a crucial task in signal processing with wide-ranging applications. Real-world signals often suffer from temporal shifts and scaling, leading to errors in raw data classification. This paper presents a novel Deep Learning-based approach for Multiple Time Series Alignment (MTSA). While existing methods mainly focus on Multiple Sequence Alignment (MSA) for biological sequences, there is a notable lack of alignment techniques for numerical time series. Traditional methods also typically address pairwise alignment, whereas our approach aligns all signals simultaneously, improving both alignment efficiency and computational speed. By decomposing to piece-wise linear sections, we introduce varying complexity into the warping function while ensuring compliance with three key constraints: boundary, monotonicity, and continuity conditions. We propose a deep convolutional network with a novel loss function that addresses key limitations of Dynamic Time Warping (DTW). Experiments on the UCR Archive 2018, involving 129 time series datasets, show that our method significantly enhances classification accuracy, warping average, and runtime efficiency across most datasets.

Keywords: Multiple Time Series Alignment, Dynamic Time Warping, Warping Function, Neural Network, Loss Function

1 Introduction

Multiple Sequence Alignment (MSA) and Multiple Time Series Alignment (MTSA) are essential in machine learning, data analysis, and bioinformatics, both aiming to align multiple inputs to identify patterns. The key difference lies in the data type: MSA aligns symbolic, discrete sequences like DNA, RNA, or proteins, while MTSA aligns continuous numerical signals, such as time series representing temporal or spatial measurements.

Both MSA and MTSA are commonly performed via successive pairwise alignments. While effective, this approach is computationally intensive, particularly for MTSA, where the numerical nature of data significantly increases complexity. Consequently, MTSA has received less attention in the literature compared to MSA. Our paper addresses this gap by introducing a true multiple alignment algorithm for MTSA that avoids repeated pairwise alignments, improving computational efficiency. Drawing on methodological parallels with MSA, we also review existing MSA strategies.

The problem involves aligning a set of time series with arbitrary lengths. Due to its importance and wide applications, various approaches have been proposed for MSA. At the heart of these methods is Dynamic Time Warping (DTW), the most widely used technique for signal alignment. In the following subsections, we present various applications of MSA and methods grounded in DTW, then we proceed to recent approaches.

1.1 Applications

The applications of MSA and MTSA can be categorized as follows:

- *Classification:* Time series classification is complicated by temporal and amplitude variations. Therefore, pre-warping can improve accuracy, as shown in our experiments. Traditional approaches combine DTW with Nearest Neighbor (NN) [1–3], but are computationally intensive. Nearest Centroid (NC) method reduces cost by aligning test samples to class representatives [4], often chosen using Dynamic Barycenter Averaging (DBA) [5]. Instead, we apply MTSA algorithms, achieving superior quality and performance
- *Human Activity Recognition:* HAR is a specialized classification task involving motion signals, widely used in surveillance, healthcare and assistive robotics. Here signal alignment is crucial due individual variations in speed and phase. Several warping-based methods for HAR have been proposed in [1, 6–10].
- *Biological Signal Analysis:* ECG, EEG, EMG, and PPG are key signals in intelligent health monitoring. Due to signal variability and limited labeled data, unsupervised warping methods are crucial. DTW has been used for sub-pattern prediction [11], noise reduction [12, 13], and neural network-based approximation for EEG signals [14].

Recently DTW and alignment methods have also been used for video alignment [11, 15, 16], time series forecasting [17, 18] and anomaly detection [19]. While there are numerous other applications, we omit them for the sake of brevity.

1.2 Classical Approaches

MSA is widely used in genomics, particularly for protein sequence analysis, leading to the development of numerous methods in this field. ClustalW [20], the first method discussed, builds a guide tree from pairwise alignments based on Progressive Alignment, assuming similar aligned signals can be merged. This iterative approach requires homogeneous signals, such as motion or ECG data.

Hidden Markov Models (HMM) are used for MSA in studies [21–23]. In [24], an unsupervised approach models time series as non-uniform samples from a latent trace, addressing local rescaling and noise. MTSA alignment is performed using DTW between each signal and the latent trace. Notably, [24] is one of the few works directly addressing numerical time series in MTSA.

In all the aforementioned works, Multiple Alignment is achieved through successive pairwise alignments. Additionally, all these methods utilize DTW for time series alignment. DTW computes the optimal alignment between two time series by allowing non-linear temporal variations. It constructs a cost matrix of pairwise distances and employs dynamic programming to identify the warping path that minimizes cumulative distance, subject to temporal constraints. For further details, readers are referred to [25].

Despite its effectiveness, DTW has several limitations. Its *polynomial computational complexity* makes it impractical for large datasets, leading to the development of approximate methods that reduce complexity to linear time while almost preserve alignment quality. DTW also suffers from *singularity*, where vertical-axis differences lead to one-to-many point mappings, distorting alignment. This can be mitigated by incorporating local shape information through shape descriptors [2] or by extracting relevant features via neural networks prior to warping [26]. Additionally, DTW is *non-differentiable*, limiting its integration into neural network training. To address this, differentiable variants like Soft-DTW [27] have been introduced.

1.3 Enhanced DTW-Based Approaches

In an attempt to address the limitations of DTW, several alternative methods have been proposed:

- *Generalized Time Warping* [7]: GTW addresses the polynomial complexity of DTW by introducing a linear-time algorithm that models the warping path as a linear combination of basis functions.
- *Trainable Time Warping* [28]: TTW enhances warping by operating in the continuous time domain with convolutional kernels, offering better performance for complex warpings.
- *Neural Time Warping* [9]: NTW relaxes the original DTW optimization problem to a continuous convex problem and finds the solution using a neural network.

Both TTW and NTW serve as approximations of the original DTW problem. Additionally, studies [29, 30] introduce modifications to DTW to enhance its effectiveness in time series classification.

1.4 Deep Learning Approaches

Integrating deep neural networks, such as Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN), into time series alignment offers substantial advantages due to their architectural flexibility, customizable loss functions, and tunable hyperparameters. Their capacity to extract meaningful features helps address challenges like singularity in DTW. Here, we review recent deep learning approaches for signal alignment.

- *Sequence Transformer Network (STN)* [31]: STN, built on CNN, enables simple translations and scalings in both time and amplitude domains, providing a powerful deep learning-based tool for time series alignment.
- *Temporal Transformer Network (TTN)* [8]: TTN is a supervised warping module placed before a classifier to reduce intra-class variability and increase inter-class separation.
- *DeepFRC* [32]: It incorporates an alignment module that learns time warping functions via elastic function registration and a learnable basis representation module for dimensionality reduction on aligned data.
- *Regularization-free Diffeomorphic Temporal Alignment Nets (RF-DTAN)* [33]: This paper introduces a novel architecture that utilizes diffeomorphic transformations for joint alignment and averaging of time series, preserving the inherent temporal structures while improving alignment quality.
- *Deep Attentive Time Warping (DATW)* [34]: The authors propose a neural network model for task-adaptive time warping using the attention model, to predict all local correspondences between two time series. The method has two learning stages: pre-training and contrastive learning.

1.5 Contributions

We propose an efficient *multiple alignment* framework that improves both the speed and accuracy of time series alignment compared to existing approaches. Moreover, as discussed previously, methods such as NTW [9] and Soft-DTW [27] are differentiable approximations of DTW. In contrast, we introduce a more effective method that addresses fundamental limitations of DTW by introducing a novel cost function. This leads to more flexible, accurate alignments. In summary, we have made the following contributions:

- *Linear Inference Complexity*: Our model achieves linear inference time, addressing the scalability limitations of previous MSA/MTSA methods with higher-order complexity.

- *Grouped MTSA Algorithm*: We introduce a grouped alignment approach that avoids redundant pairwise comparisons, improving both efficiency and scalability for MTSA through a novel training and testing procedure.
- *Holistic Loss Function for Deep Alignment*: We propose a novel loss that captures overall similarity between signals with a cosine similarity-based metric, going beyond point-to-point proximity, enabling more accurate and flexible alignment in deep models.
- *Stability-Promoting Penalization Terms*: We introduce carefully designed penalties to encourage stable and reliable alignment outcomes.
- *Empirical Validation Across Datasets*: We demonstrate that our method improves classification performance on most of datasets in the UCR Archive 2018, highlighting its practical utility.

2 The Proposed Method

In this section, we first present the necessary background definitions and formally introduce the MTSA problem. We then provide a detailed description of our proposed method, highlighting its key innovation s.

2.1 Background and Problem Definition

Warping Path: Consider two time series *X* and *Y* with lengths *N* and *M*, respectively. The warping path, denoted as *P*, is a sequence with length $L \in \mathrm{N}$ defined as follows:

$$P = (p_1, \ldots, p_l) \tag{1}$$

In Equation 1 for $l \in [1:L]$ we have $p_l = (n_l, m_l) \in [1:N] \times [1:M]$ Clearly $L = \max(N, M)$ and $p_l = (n_l, m_l)$ indicates that index n_l from *X* is warped to index m_l from *Y*. This warping path contains all essential information for aligning the two signals. Typically, three *warping constraints* are considered:

- *Boundary condition*: $p_1 = (1,1)$ and $p_L = (N, M)$. This ensures the first and last indices from the signals are warped to each other.
- *Monotonicity condition*: $n_1 \leq n_2 \leq \ldots \leq n_L$ and $m_1 \leq m_2 \leq \ldots \leq m_L$. The alignment must preserve the chronological order of the time series.
- *Continuity condition*: $p_{l+1} - p_l \in \{(1,0), (0,1), (1,1)\}$ for each $l \in [1:L]$. This prevents discontinuities between corresponding points, ensuring that each time step has at least one match in the other signal.

MTSA Problem Definition: Suppose N time series $X_1, X_2, \ldots, X_N$, where for $i \in [1:N], X_i \in \mathrm{R}^{d_i \times T_i}$ with d_i, T_i representing the dimension and length of X_i, respectively. Two models can be employed to express time warping:

- *Matrix Multiplication:* Defining warping matrices as W_i for $i \in [1:N]$, the warped form of X_i can be expressed as $W_i X_i$. One possible MSE cost function for the MTSA problem can be formulated as Equation 2:

$$J_{MTSA1}(\{W_i\}) = \sum_{i=1}^{N}\sum_{j=1}^{N} \left\| W_i X_i - W_j X_j \right\|^2 \tag{2}$$

- *Function Composition:* Utilizing warping functions τ_i for $i \in [1:N]$, the warped form of X_i is $X_i \circ \tau_i = X_i(\tau_i(t))$ and the associated cost function can be expressed as Equation 3:

$$J_{MTSA2}(\{\tau_i\}) = \sum_{i=1}^{N}\sum_{j=1}^{N} \left\| X_i(\tau_i(t)) - X_j(\tau_j(t)) \right\|^2 \tag{3}$$

2.2 Warping Function and Constraints

To generalize linear warping $\tau(t) = at + b$ and model complex temporal alignments, we propose a *piecewise linear warping function* as follows:

- *Segmentation*: The time axis is divided into K consecutive intervals with durations $t_1, t_2, \ldots, t_K$. The k-th interval spans $\left[\sum_{i=1}^{k-1} t_i, \sum_{i=1}^{k} t_i\right)$.
- *Slopes*: The k-th interval is assigned a slope a_k, controlling how quickly time progresses within that segment.
- *Warping Function*: The function $\tau(t)$ is continuous and linear within each interval, defined by $\tau(t) = \sum_{i=1}^{k-1} a_i t_i + a_k \left(t - \sum_{i=1}^{k-1} t_i\right)$, for $t \in \left[\sum_{i=1}^{k-1} t_i, \sum_{i=1}^{k} t_i\right)$ in the k-th interval.
- *Neural Network Output*: A neural network predicts $2K$ non-negative parameters $\{a_1, \ldots, a_K\}$ and $\{t_1, \ldots, t_K\}$, enabling the function to adapt its shape based on input data. The higher K increases non-linearity, allowing for a more flexible alignment.

The proposed warping function, illustrated in Fig. 1 and defined in Equation 4, satisfies three fundamental warping constraints:

- *Boundary condition*: It is evident that $\tau(0)=0$. Additionally, we enforce $\sum_{k=1}^{K} t_k = T$, where T is the length of the target warped signal.
- *Monotonicity condition*: This condition holds if $a_k \geq 0$ for $k \in [1:K]$. Ensuring non-negative slopes guarantees a monotonically increasing warping function.
- *Continuity condition*: The function $\tau(t)$ is continuous, thus satisfying the continuity constraint.

$$\tau(t)=\begin{cases} a_1 t & t<t_1 \\ a_1 t_1 + a_2(t-t_1) & t_1 \leq t < t_1+t_2 \\ \dots & \dots \\ \sum_{k=1}^{K-1} a_k t_k + a_K (t-\sum_{k=1}^{K-1} t_k) & \sum_{k=1}^{K-1} t_k \leq t < \sum_{k=1}^{K} t_k \end{cases} \tag{4}$$

2.3 Non-differentiability Problem

Consider a neural network is trained to implement the warping function $\tau(\cdot)$, and let signal X with length T be inputted to the network. The warped signal is obtained as $X_{warp} = X(\tau(\cdot))$. Consequently, $X(\tau(t))$ should be calculated for each $t \in [1,T]$. However, if $\tau(t)$ is not an integer, standard (*hard*) warping approximates it to the nearest integer since X is defined only at discrete time steps. This makes the loss function non-differentiable, as small changes in time (t_k) or amplitude (a_k) parameters may result in non-integer $\tau(t)$, causing $X(\tau(t))$ and the loss function to be undefined. Consequently, gradient-based optimization cannot be applied.

To solve this, *soft warping* is utilized, allowing $\tau(t)$ to be a floating-point value. The warped signal X_{warp} is then computed using interpolation. This is modeled through matrix multiplication (Equation 2), where the warping matrix W contains values in the range [0,1].

2.4 Neural Network Structure

The overall structure of the neural network is illustrated in Fig. 2. The input time series $X_1(t), X_2(t), \dots, X_N(t)$ are assumed to have the same length at this stage; considerations for different-length time series will be addressed later. The primary network is a CNN with an input, three convolutional, a flatten and two dense layers.

- *Input Layer*: Receives $X_i(t)$ from the dataset and passes it to the first convolutional layer.

- *Convolutional Layers*: Comprise multiple convolutional kernels and pooling layers to extract features.
- *Flatten Layer*: Converts the final convolutional layer's output into a vector proportional to the input time series length.
- *Parallel Dense Layers*: Two layers generate the warping function parameters $\{a_1, a_2, a_3, a_4\}$ and $\{t_1, t_2, t_3, t_4\}$, as shown in Fig. 1 for $K = 4$.

From these outputs a warping function is implemented, and a warping matrix W_i is calculated using the soft warping concept. The warped input $X_{i,warp}(t)$ is obtained by multiplying X_i with W_i and is applied to both the loss function and the input dataset blocks.

Two key contributions related to the neural network include the *loss function block* and the *training and testing procedure*, which will be discussed in the following subsections.

2.5 Loss Function

As discussed in Section 1, DTW faces issues like computational complexity and singularity. To address *singularity*, we propose two solutions: First, using convolutional kernels in CNNs for feature extraction, allowing local patterns at each temporal point to influence adjacent points, creating relationships between them. Second, instead of relying on traditional DTW algorithms with MSE loss functions, which can cause singularity due to their point-wise nature, we implement a more robust loss function that captures the overall similarity between two signals, rather than just point-to-point proximity. This function can be split into *main* and *penalization* parts, which will be deliberated in this section.

The Main Part: It must accommodate small to moderate scalings and shifts in the temporal domain without correcting amplitude. So, when two signals are multiples of each other, the loss function should reach its minimum. The approach is to apply the *inner product* of the two signals. For two arbitrary 1-dimensional signals *X* and *Y* (vectors), the *Cosine Similarity* function is defined as follows:

$$S_X(X, Y) = \frac{\langle X, Y \rangle}{\max\left\{\|X\|_2 \|Y\|_2, \grave{o}\right\}} \qquad (5)$$

Here, $\|\cdot\|_2$ denotes the Euclidean norm, and $\grave{o}$ is a small positive constant to prevent division by zero. Cosine similarity ranges from [-1, 1], where 1 signifies codirectional signals, 0 indicates orthogonal signals, and -1 represents contradirectional signals. To achieve smoother results, we use a quadratic form of cosine similarity while preserving its sign. This is because both orthogonal and contradirectional signals are undesirable, and we need codirectional signals. Consequently, the loss function in Equation 6 is defined using the signed square form of cosine similarity.

$$L(X,Y)=1-S_C(X,Y)^2\,sign(S_C) \qquad (6)$$

Finally, the main loss function between two arbitrary signals X and Y is introduced as Equation 7:

$$L_{main}(X,Y)=L(X_{warp},Y) \qquad (7)$$

The main loss function in Equation 7 is similar to Equation 6, only the first signal (X) is warped and then its cosine similarity with the second signal (Y) is measured.

If the signals have dimensions greater than one, each row is treated as an individual vector. Cosine similarity is then calculated between corresponding rows using Equation 5, resulting in a vector as the main loss function in Equation 7, with a size equal to the signal dimensions. To obtain a specific loss function, the average value of the elements in this vector is computed.

The Penalization Part: If $a_i=1$ for all $i\in[1:K]$, the warping function becomes the identity, implying no change to the signal. Since signals in the dataset are assumed to be homogeneous with minimal discrepancies, the values of $\{a_1,\dots,a_K\}$ should stay close to 1. To encourage this, two *penalization terms* are added to the loss function. Suppose x is a measure of the mean amplitude of $\{a_1,\dots,a_K\}$. We define two functions on x:

- $f_1(x)=(x-1)^2$: Encourages x to be around 1 and penalizes x for values far larger than 1.
- $f_2(x)=1/(x^2+\grave{o})$: Prevents x from going too close to zero. Here, $\grave{o}$ is a small positive constant.

The combination of these two functions can be expressed as Equation 8, and Fig. 3 illustrates its graphical curve.

$$f(x)=(x-1)^2+\frac{1}{x^2+0.1} \qquad (8)$$

Based on Fig. 3, the function in Equation 8 can serve as an effective penalization term. Building on this prototype, we define the following penalization function:

$$L_{pen.}(a_1,\ldots,a_K)=\sum_{k=1}^{K}(a_k-1)^2+\lambda_1\frac{1}{\frac{1}{K}\sum_{k=1}^{K}a_k^2+0.1} \tag{9}$$

The Final Loss Function: Combining Equation 9 with Equation 6, the loss function for an input time series *X* can be expressed as Equation 10:

$$\begin{aligned} L_{final}(X,Y)&=L_{main}(X,Y)+\lambda_2 L_{pen.}(a_1,\ldots,a_K)= \\ &1-S_C(X_{warp},Y)^2 sign(S_C)+\lambda_2\left(\sum_{k=1}^{K}(a_k-1)^2+\lambda_1\frac{1}{\frac{1}{K}\sum_{k=1}^{K}a_k^2+0.1}\right)\end{aligned} \tag{10}$$

In Equations 9 and 10, λ_1 and λ_2 are hyper-parameters controlling the penalization terms, and a_k for $k\in[1:K]$ are the amplitude outputs corresponding to the input *X*. The main loss function $L_{main}(X,Y)$, is computed between the warped input signal X_{warp} and the second signal *Y*. For two signals *X* and *Y*, the neural network can warp the first signal *X* to align with *Y* using Equation 10. For more than two time series, the problem becomes MTSA, discussed in the next subsection.

2.6 Training and Testing Procedure

Here, we explain how our framework extends to the multiple case for the MTSA problem. Consider Fig. 2, where the signals in the input dataset X_i for $i\in[1:N]$ have the same length *T*. If their lengths differ, a pre-processing stage will equalize them. Below is the proposed algorithm for the training procedure:

1. Apply each time series X_i to the network input.
2. Obtain amplitude parameters $\{a_1,a_2,a_3,a_4\}$ and time parameters $\{t_1,t_2,t_3,t_4\}$ from the network.
3. Utilize the warper block to generate the warping matrix based on these parameters and multiply it with the input time series to construct $X_{i,warp}$.
4. The loss function block calculates the average final loss between $X_{i,warp}$ and each of the other $N-1$ signals according to Equation 10.
5. Replace the original X_i with its warped version $X_{i,warp}$.

6. Repeat steps 1-5 for all *N* signals, completing one epoch of training.
7. Perform the required number of epochs to gradually align signals to each other.

Substituting signals with their warped versions is essential in our MTSA framework. However, early in training, the network may lack meaningful warpings. Delaying substitution until the model learns more relevant information ensures stable and informed dataset updates.

Ultimately, the network aligns *N* input signals, enabling accurate warping of homogeneous *test* time series. During testing (illustrated in Fig. 2), the process remains the same except for omitting the loss function block. The input test signal X_i is processed by the network, producing the warped test signal $X_{i,warp}$ via the warper block.

A key benefit of using deep neural networks for time series alignment is the elimination of backpropagation during testing. Unlike conventional methods such as DTW, which require repeated optimization for each alignment, our approach uses a parameterized network that learns to align signals efficiently.

3 Experiments

Four series of experiments are presented in this paper. The first addresses the MTSA problem by aligning test signals to training signals. The second explores warped averaging as a key MTSA application, highlighting notable cases to evaluate the method's performance. The third involves a classification test on 90 datasets, reporting accuracy for a Nearest Neighbor classifier. The fourth validates the method's superiority by measuring classification rate and error using a deep ResNet classifier. Experiments were conducted on the UCR Time Series Classification Archive [35], a widely recognized benchmark in time series analysis, as it is commonly employed in the majority of existing literature for evaluating alignment and classification methodologies.

Missing Values and Varying Lengths: The UCR Archive includes 11 datasets with *variable time series lengths*, which require pre-processing to equalize them. Following [36], we adjust each time series based on the average sequence length: longer series are shortened by randomly removing time steps, while shorter ones are extended by inserting values computed as the average of random points and their neighbors. This approach preserves shape and is more efficient than uniform stretching, which requires recomputing all values. *Missing values* are handled via linear interpolation, with boundary cases addressed through nearest-value extrapolation to maintain continuity without discarding samples.

Experiment Setup: The CNN consists of three layers with filter sizes of 13, 7 and 3, and filter counts of 128, 64 and 32, respectively. Each convolutional layer is followed by an average pooling layer (stride 1, sizes 6, 4 and 2). After the third layer, the tensor is flattened and processed by two *parallel* dense layers, each with 4 output neurons representing $\{a_1, a_2, a_3, a_4\}$ and $\{t_1, t_2, t_3, t_4\}$. ReLU activation functions ensure non-negative, unbounded outputs for a and t values. The hyperparameters λ_1 and λ_2 in Equation 10 are set to 0.5 for most datasets. Although optimizing them individually could improve results, we avoided this due to its time-intensive nature. The learning rate is fixed at 10^{-3}. Training runs for 25 epochs, with checkpoints saved every 5 epochs to account for potential early stopping benefits. The best model is chosen based on validation accuracy. The implementation uses the PyTorch library.

3.1 The Multiple Time Series Alignment (MTSA)

A key application of MTSA is computing a *warped average* to represent a set of signals, as a simple arithmetic average cannot handle temporal shifts or scale variations. DBA [5], a robust MTSA method, iteratively uses DTW to align signals with an evolving average. Also, RF-DTAN [33] is a novel deep-learning based approach for MTSA. In this study, both DBA and RF-DTAN are used as baselines for MTSA (this section) and warped averaging (the next section) to demonstrate the advantages of our proposed time series alignment approach.

For each dataset, signals with the same label are inputted into the model to ensure homogeneity. Standard UCR dataset train-test splits are used. The goal is to optimally align 5 test signals with their corresponding training signals. Fig. 4 illustrates the results for various datasets and labels. For each test signal (red), generating its warped counterpart (green) involves solving an MTSA problem to align it with a set of training signals (gray). In cases like *"Plane: 4"* and *"Trace: 3"*, simple linear transformations are insufficient, requiring more complex non-linear warpings for accurate alignment. Once the warper network is trained, the MTSA problem is solved by passing the test signal through the network, ensuring linear computational complexity relative to signal length. Notably, inference time is unaffected by the number of training signals, making the method scalable for large datasets. A major advantage of deep neural networks is the decoupling of training time (a one-time process) from test time.

We compare the computation times of our method against DBA and RF-DTAN for generating warped signal averages. While quality metrics are discussed in Subsection 3.2, this section focuses on timing. As shown in Table 1, our model is over twice as fast as DBA and substantially faster than RF-DTAN. Fig. 5 illustrates performance across all UCR datasets, with our method outperforming DBA in over 82% of cases and consistently surpassing RF-DTAN. Notably, it reduces DBA’s processing time from 258 to 59 seconds, achieving more than 4-fold improvement.

3.2 Representative and Warped Averaging

In this section, we provide visual comparisons demonstrating the advantages of our approach over the DBA and RF-DTAN algorithms in computing the warped average signal and effectively addressing various challenges.

Overall Comparison: An overall test on the GunPoint dataset evaluates our method's performance, as shown in Fig. 6. Fig. 6(a) displays label 1 signals with different averaging methods. Fig. 6(b) shows warped signals using our method and their average. Fig. 6(c),(d) present the same for label 2. The results highlight that the simple average fails to capture slightly complex trends, particularly for label 2, while DBA introduces unwanted spikes and RF-DTAN does not seem as a proper average. In contrast, our method aligns signals effectively, producing a warped average that preserves the trend of signals and serves as a representative for each class.

Preserve Signal Shapes: This is crucial in warped averaging, especially for challenging datasets like Trace. Simple averaging fails to capture the true shape of signals, as shown in Fig. 7. While DBA improves the results, RF-DTAN and our approach effectively compensate for signal shifts by applying appropriate multiple warping. This generates a warped average with reduced variations and better representation of the underlying trend compared to DBA.

Alignment of Peaks: The InsectWingbeatSound dataset contains signals with sequences of unaligned peaks, making alignment and trend extraction very challenging. Fig. 8 demonstrates that both simple averaging and DBA fail to preserve the sequence of peaks, particularly smaller ones. In contrast, RF-DTAN and our proposed method successfully preserve the peak sequences. Notably, our method produces cleaner and more representative average signals.

Signal Shifts: Time warping effectively compensates for temporal shifts in signals with similar shapes. As demonstrated in Fig. 9, our method successfully removes temporal displacements, resulting in warped signals that produce a more accurate average trend compared to other approaches.

Noisy Environments: Extracting signal shapes from datasets with high variation and noise is challenging. However, as shown in Fig. 10 on the SyntheticControl and CBF datasets, our method effectively aligns signals and extracts a meaningful representative for the time series set, even under noisy conditions.

Outlier Signals: Rare signals with peaks at specific temporal points are considered outlier trends and should be excluded from the representative signal. As shown in Fig. 11 on the MoteStrain dataset, while other methods reflect local peaks, our model produces a representative signal that captures the overall trend without these outliers.

3.3 The Comprehensive Classification Test

This section and the next aim to show how our proposed warper network enhances *classification accuracy*. We employ the nearest neighbor (NN) classifier, and evaluate accuracy across datasets under several conditions: a basic NN classifier, NN combined with DTW and DBA [5] (classical approaches) and NN with RF-DTAN [33], DATW [34] and our method (deep learning-based approaches).

In the DTW+NN classifier, the Euclidean distance is replaced with DTW distance, requiring DTW computation between the test sample and all training signals. In the DBA approach, the warped average of training signals is computed for each class, and test samples are assigned to the class whose representative has the smallest DTW distance. RF-DTAN and DATW combinations with NN are implemented as described in their original publications.

In our approach, a neural network is trained for each class using specified parameters. Training is repeated with multiple random initializations, and the best model is selected based on validation accuracy. The final model's performance is evaluated on the test dataset.

After training on a dataset, each test signal is processed through all class-specific warpers. The error is measured between the warped test signal and *the average* of all warped training signals for each class (warped by their corresponding class warper) using Equation 6. The test signal is assigned to the class whose warper produces the smallest error.

A limitation of our approach is the requirement to train as many models as there are classes in a dataset, making it less practical for datasets with numerous classes. Due to this and resource constraints, we performed classification tests on 90 UCR Archive datasets. Experiments were conducted on a system with an 8-core CPU and 64 GB RAM. Since DATW is optimized for GPU and requires several days per dataset on CPU, we used published results from [34], adding a few supplementary datasets. RF-DTAN was run on 90 datasets with an average runtime of 65 minutes per dataset, while our method achieved an average of just 4 minutes per dataset.

The results are presented in Table 2. This table demonstrates that our method on average improves baseline results by **6.1%**, DTW+NN by **3.1%**, DBA+NN by **7.5%**, DATW by **3%** and RF-DTAN by **6.2%**. The last row in Table 2 shows the Mean Per Class Error (MPCE) introduced by [37], which is defined as Equation 11.

$$MPCE = \frac{1}{K}\sum_{k=1}^{K}\frac{1-Acc_k}{Number\ of\ \ classes} \tag{11}$$

In Equation 11, Acc_k is the classification accuracy in the k th dataset, and K is the number of datasets. MPCE measures the expected error rate per class across all datasets. According to Table 2, our method reduces the MPCE by **24.6%** compared to NN, **17.5%** compared to DTW+NN, **28.8%** compared to DBA+NN, **18.9%** compared to DATW and **27.2%** compared to RF-DTAN. Thus, on average, it exhibits better classification accuracy per class for these 90 datasets.

Error Analysis: The final column of Table 2 presents cosine similarity-based loss between training signals before and after training. As some UCR datasets are manually aligned, warping may not always improve alignment, and in some cases, suitable warping functions may not exist. These limitations are reflected in the loss differences between original and warped signals. Datasets are sorted by the degree of loss reduction; those at the top show greater improvements in both loss and accuracy over the NN baseline. Conversely, lower-ranked datasets (e.g., OliveOil, Fungi, and Meat) are already well-aligned, making warping less effective and a simple NN more suitable. Minor accuracy drops in some datasets are attributed to using uniform hyperparameters, which could be mitigated through dataset-specific fine-tuning.

Statistical Significance Test: The results of *Wilcoxon signed-rank* test are presented in Table 3, which confirm that our method achieves a statistically significant improvement over four baselines: NN, DTW+NN, DBA+NN and RF-DTAN (with the p-value lower than 0.05). Compared to DATW, although our method achieves a higher average accuracy, the difference is not statistically significant (p-value is 0.275) due to limited number of datasets. However, the total processing time of our model is significantly lower than DATW (4 minutes compared to several days per dataset).

Finally, Fig. 12 illustrates the wins and losses of our model compared to NN, DTW+NN, RF-DTAN and DATW baselines. In each plot, blue points indicate wins and red points represent losses. As shown, our model outperforms NN on 65 out of 90 datasets, with 15 losses; against DTW+NN, it records 50 wins and 33 losses; against RF-DTAN, 72 wins and 17 losses; and against DATW, 27 wins and 17 losses. These results highlight the effectiveness of our approach across all baselines.

3.4 Deep Network Classification

After evaluating our method's effectiveness in enhancing the accuracy of a simple NN classifier, this section examines its performance with a more advanced and complex classifier.

In [38] deep learning methods for time series classification are explored, identifying *ResNet* [37] as the best-performing model among nine top-rated approaches for UCR Archive datasets. Our method is not an alternative to ResNet but can serve as a pre-stage warper to improve the accuracy. To demonstrate this, we randomly selected 30 datasets from the previous 90 (due to computational constraints) and trained the ResNet classifier for 1500 epochs, as recommended in [37]. Each dataset was tested twice: once in its original form and once after warping, where each test signal was warped using the model that produced the least error.

Table 4 presents the results, showing percentage improvements in test loss average and variance for the selected datasets. These values are computed from epoch 300 to 1500 to exclude high initial variations. The results indicate a **33%** improvement in average loss and a **54%** reduction in variance when incorporating our warper stage. Additionally, Table 4 reports final test accuracies, revealing a **2.5%** average accuracy improvement and a **22.7%** reduction in MPCE. Notably, our approach is significantly faster than ResNet, ensuring that its integration does not introduce noticeable computational overhead.

4 Discussion

The proposed innovative MTSA algorithm enables scalable multiple sequence alignment, offering superior speed and alignment quality compared to baseline models. By employing a novel and efficient loss metric, incorporating appropriate regularization terms, and utilizing an optimized learning procedure, our network effectively learns class-specific patterns from training signals and accurately identifies them in test signals. However, as noted in the Experiments section, a primary limitation of our approach is the need to train a separate neural network for each class in the dataset. Consequently, the method may be less practical for datasets with a large number of classes.

Incorporating time warping into time series analysis opens possibilities for applications such as forecasting and anomaly detection, making these promising directions for future work. Additionally, extending our approach to multi-dimensional time series could enable its use in tasks such as video alignment.

5 Conclusion

We presented a novel deep learning-based framework for MTSA, addressing a largely overlooked problem in the literature. Unlike traditional MSA methods that relied on pairwise alignments, leading to high computational complexity, in our approach a grouped multiple alignment algorithm was introduced that jointly aligned all signals. Additionally, complex non-linear warpings were decomposed into simple linear sections and a penalization term was added to the loss function, ensuring a general time warping that adheres to three essential constraints. By optimizing loss functions and training procedures, our method achieved promising results in both time series classification and warped averaging.

References

[1] Folgado, D., Barandas, M., Matias, R., *et al.*: "Time alignment measurement for time series". *Pattern Recognition* **81**, 268–279 (2018) https://doi.org/10.1016/j.patcog.2018.04.003

[2] Zhao, J., Itti, L.: "ShapeDTW: shape dynamic time warping". *Pattern Recognition* **74**(3) (2017) https://doi.org/10.1016/j.patcog.2017.09.020

[3] Cui, L., Zhang, Q., Shi, Y., *et al.*: "A method for satellite time series anomaly detection based on fast-DTW and improved-KNN". *Chinese Journal of Aeronautics* **36**(2), 149–159 (2023) https://doi.org/10.1016/j.cja.2022.05.001

[4] Petitjean, F., Forestier, G., Webb, G., et al.: "Dynamic time warping averaging of time series allows faster and more accurate classification". *IEEE International Conference on Data Mining*, 470–479 (2014) https://doi.org/10.1109/ICDM.2014.27

[5] Shi, K., Qin, H., Li, S., *et al.*: "Dynamic barycenter averaging Kernel in RBF networks for time series classification". *IEEE Access* **7**, 47564–47576 (2019) https://doi.org/10.1109/ACCESS.2019.2910017

[6] Ghodsi, S., Mohammadzade, H., Korki, E.: "Simultaneous joint and object trajectory templates for human activity recognition from 3-D data". *Journal of Visual Communication and Image Representation* **55**, 729–741 (2018) https://doi.org/10.1016/j.jvcir.2018.08.001

[7] Zhou, F., Torre, F.: "Generalized time warping for multi-modal alignment of human motion". *IEEE Conference on Computer Vision and Pattern Recognition*, 1282–1289 (2012) https://doi.org/10.1109/CVPR.2012.6247812

[8] Lohit, S., Wang, Q., Turaga, P.: "Temporal transformer networks: joint learning of invariant and discriminative time warping". *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12418–12427 (2019) https://doi.org/10.48550/arXiv.1906.05947

[9] Kawano, K., Kutsuna, T., Koide, S.: "Neural time warping for multiple sequence alignment". *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3837–3841 (2020) https://doi.org/10.1109/ICASSP40776.2020.9054121

[10] Xing, J., Li, H.: "Study about football action recognition method based on deep learning and improved dynamic time warping algorithm". *Mobile Information Systems* **2022** (2022) https://doi.org/10.1155/2022/3861620

[11] Hooi, B., Liu, S., Smailagic, A., et al.: “BEATLEX: summarizing and forecasting time series with patterns”. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 3–19 (2017) https://doi.org/10.1007/978-3-319-71246-8 1

[12] Malghan, P., Hota, M.K.: “Grasshopper optimization algorithm based improved variational mode decomposition technique for muscle artifact removal in ECG using dynamic time warping”. *Biomedical Signal Processing and Control* **73** (2022) https://doi.org/10.1016/j.bspc.2021.103437

[13] Souriau, R., Fontecave-Jallon, J., Rivet, B.: “Fetal ECG denoising using dynamic time warping template subtraction”. *44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 4978–4981 (2022) https://doi.org/10.1109/EMBC48229.2022.9871318

[14] Lerogeron, H., Picot-Clemente, R., Rakotomamonjy, A., *et al.*: “Approximating dynamic time warping with a convolutional neural network on EEG data”. *Pattern Recognition Letters* **171**, 162–169 (2023) https://doi.org/10.1016/j.patrec.2023.05.012

[15] Fakhfour, N., ShahverdiKondori, M., Hashembeiki, S., *et al.*: “Video alignment using unsupervised learning of local and global features”. *arXiv preprint arXiv*: 2304.06841 (2024) https://doi.org/10.21203/rs.3.rs-3457319/v1

[16] Haresh, S., Kumar, S., Coskun, H., et al.: “Learning by aligning videos in time”. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5548–5558 (2021) https://doi.org/10.48550/arXiv.2103.17260

[17] Lin, Y., Koprinska, I., Rana, M.: “SpringNet: transformer and spring DTW for time series forecasting”. *Neural Information Processing*, 616–628 (2020) https://doi.org/10.1007/978-3-030-63836-8 51

[18] Tao, Z., Xu, Q., Liu, X., *et al.*: “An integrated approach implementing sliding window and DTW distance for time series forecasting tasks”. *Applied Intelligence* **53**(17), 894–903 (2023) https://doi.org/10.1007/s10489-023-04590-9

[19] Abilasha, S., Bhadra, S.: “Warping resilient robust anomaly detection for multivariate time series”. *Machine Learning* **114**(2) (2025) https://doi.org/10.1007/s10994-024-06689-7

[20] Thompson, J., Higgins, D., Gibson, T.: “CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice”. *Nucleic Acids Research* **22**(22), 4673–4680 (1994) https://doi.org/10.1093/nar/22.22.4673

[21] Zhang, C., Zheng, W., Mortuza, S., *et al.*: “DeepMSA: constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant-homology proteins”. *Bioinformatics* **36**(7), 2105–2112 (2020) https://doi.org/10.1093/bioinformatics/btz863

[22] Park, M., Warnow, T.: “HMMerge: an ensemble method for multiple sequence alignment”. *Bioinformatics Advances* **3**(1), 052 (2023) https://doi.org/10.1093/bioadv/vbad052

[23] Shen, C., Park, M., Warnow, T.: "WITCH: improved multiple sequence alignment through weighted consensus Hidden Markov Model alignment". *Journal of Computational Biology* **29**(8) (2022) https://doi.org/10.1089/cmb.2021.0585

[24] Listgarten, T., Neal, M., Emili, A.: "Multiple alignment of continuous time series". *Advances in Neural Information Processing Systems* **17** (2004)

[25] Mller, M.: "Dynamic time warping". *Information retrieval for music and motion* (2007)

[26] Wu, X., Kimura, A., Iwana, B., et al.: "End-to-end local representation learning for online signature verification". *International Conference on Document Analysis and Recognition (ICDAR)*, 1103–1110 (2019) https://doi.org/10.1109/ICDAR.2019.00179

[27] Cuturi, M., Blondel, M.: "Soft-DTW: a differentiable loss function for time-series". *Proceedings of the 34th International Conference on Machine Learning* **70**, 894–903 (2017) https://doi.org/10.48550/arXiv.1703.01541

[28] Khorram, S., Mclnnis, M., Provost, E.: "Trainable time warping: aligning time-series in the continuous time-domain". *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3502–3506 (2019) https://doi.org/10.1109/ICASSP.2019.8682322

[29] Herrmann, M., Tan, C., Webb, G.: "Parameterizing the cost function of dynamic time warping with application to time series classification". *Data Mining and Knowledge Discovery* **37**(2024–2045), 5 (2023) https://doi.org/10.1007/s10618-023-00926-8

[30] Liu, Y., Zhang, Y., Zeng, M., *et al.*: "A novel distance measure based on dynamic time warping to improve time series classification". *Information Sciences* **656**, 119921 (2024) https://doi.org/10.1016/j.ins.2023.119921

[31] Oh, J., Wang, J., Wiens, J.: "Learning to exploit invariances in clinical time-series data using sequence transformer networks". *Proceedings of the 3rd Machine Learning for Healthcare Conference* **85**, 332–347 (2018) https://doi.org/10.48550/arXiv.1808.06725

[32] Jiang, S., Hu, Y., Li, W., *et al.*: "DeepFRC: an end-to-end deep learning model for functional registration and classification". *arXiv preprint arXiv*: 2501.18116 (2025) https://doi.org/10.48550/arXiv.2501.18116

[33] Weber, S., Freifeld, O.: "Diffeomorphic temporal alignment nets for time-series joint alignment and averaging". *arXiv preprint arXiv*: 2502.06591 (2025) https://doi.org/10.48550/arXiv.2502.06591

[34] Matsuo, S., Wu, X., Atarsaikhan, G., et al.: "Deep attentive time warping". *Pattern Recognition* **136** (2023) https://doi.org/10.1016/j.patcog.2022.109201

[35] Dau, H., Keogh, E., Kamgar, K., *et al.*: "The UCR time series classification archive". *IEEE/CAA Journal of Automatica Sinica* **6**(6), 1293–1305 (2018) https://doi.org/10.48550/arXiv.1810.07758

[36] Akyash, M., Mohammadzade, H., Behroozi, H.: "Dtw-merge: a novel data augmentation technique for time series classification". *arXiv preprint arXiv*: 2103.01119 (2021) https://doi.org/10.48550/arXiv.2103.01119

[37] Wang, Z., Yan, W., Oates, T.: "Time series classification from scratch with deep neural networks: a strong baseline". *International Joint Conference on Neural Networks (IJCNN)*, 1578–1585 (2017) https://doi.org/10.1109/IJCNN.2017.7966039

[38] Fawaz, H.I., Forestier, G., Weber, J., *et al.*: "Deep learning for time series classification: a review". *Data Mining and Knowledge Discovery* **33**, 917–963 (2019) https://doi.org/10.1007/s10618-019-00619-1

6 Biographies

Alireza Nourbakhsh received B.Sc. and M.Sc. degrees in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2017 and 2019, respectively. He is currently pursuing a Ph.D. degree in electrical engineering with Digital Electronics Group, Sharif University of Technology, Tehran, Iran.

Hoda Mohammadzade is an associate professor at Sharif University of Technology, Department of Electrical Engineering. She received her BSc degree from Amirkabir University of Technology (Tehran Polytechnic), Iran, in 2004, the MSc degree from the University of Calgary, Canada, in 2007, and the PhD degree from the University of Toronto, Canada, in 2012, all in electrical engineering. Her research interests include machine learning, computer vision, biomedical signal processing, and biometrics. She has published several papers in international journals and conferences.

7 List of Captions

Fig. 1: The implemented piecewise linear warping function τ (t).

Fig. 2: The overall structure of the network including the Main Network and the Warper and Loss Function blocks.

Fig. 3: A graphical curve from the prototype penalization function f(x). This function prevents x from getting too high or too close to zero.

Fig. 4: Results of the MTSA experiment, with dataset names and labels displayed above each. In each plot, gray signals represent the warped training signals, while red signals indicate five randomly selected test signals requiring alignment. The green signals show the warped versions of the red signals, generated by our model.

Fig. 5: Scatter plots to compare the timing of our method with: (a) DBA and (b) RFDTAN.

Each point represents a label from a dataset, with points above the $y = x$ line indicating a win (blue points) and those below showing a loss (red points) for our model.

Table 1: Timing comparison for the MTSA task between our method and the DBA and RF-DTAN approaches.

Fig. 6: Results on the GunPoint dataset. (a) label 1, gray: original time series, red: simple average, green: DBA, cyan: RF-DTAN, blue: OUR warped average. (b) label 1, gray: warped time series with OUR method, blue: OUR warped average. (c) label 2, similar to (a) for colors. (d) label 2, similar to (b) for colors.

Fig. 7: Results on the Trace dataset, label 2. For details refer to Fig. 6 caption.

Fig. 8: Results on the InsectWingbeatSound dataset, (a), (b): label 2 and (c), (d): label 10. For details refer to Fig. 6 caption.

Fig. 9: (a), (b): Results on the Plane dataset, label 5. (c), (d): Results on the ECGFiveDays dataset, label 1. For details refer to Fig. 6 caption.

Fig. 10: (a), (b): Results on the SyntheticControl dataset, label 2. (c), (d): Results on the CBF dataset, label 3. For details refer to Fig. 6 caption.

Fig. 11: Results on the MoteStrain dataset. gray: original time series, red: simple average, green: DBA signal, blue: warped average with our method. (a): label 1, (b): label 2. Table 2: Classification accuracy comparison between our method and five baselines over 90 datasets of the UCR Archive.

Table 3: Results of the Wilcoxon signed-rank test to assess the statistical significance of our approach over five baselines.

Fig. 12: Scatter plots comparing our method with (a) NN, (b) DTW+NN, (c) RF-DTAN and (d) DATW. Each point represents a dataset, with points above the $y = x$ line indicating a win (blue points) and those below showing a loss (red points) for our model.

Table 4: RESNET Test Loss Average and Variance percentage improvements over epochs (after epoch 300) and Accuracy comparison for 30 datasets when a warping stage with our approach is added.

8 Figures and Tables

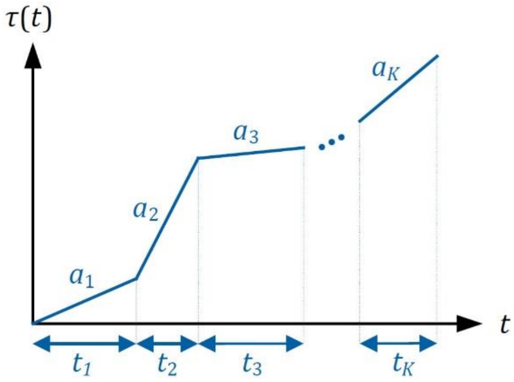


Fig. 1: The implemented piecewise linear warping function $\tau(t)$.

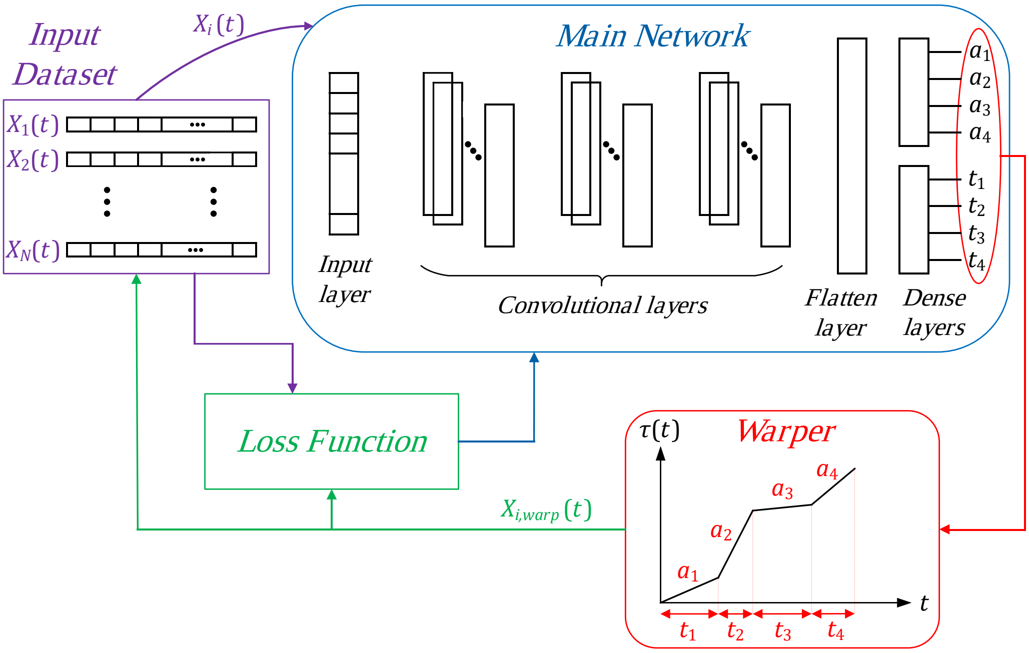


Fig. 2: The overall structure of the network including the Main Network and the Warper and Loss Function blocks.

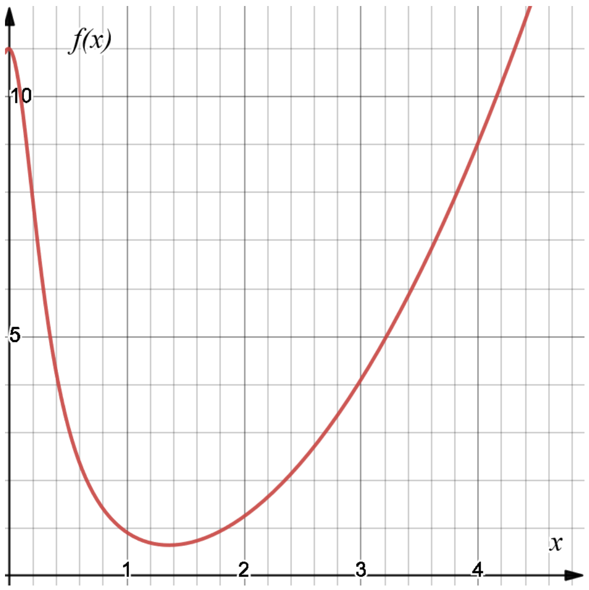


Fig. 3: A graphical curve from the prototype penalization function $f\ (x)$. This function prevents *x* from getting too high or too close to zero.

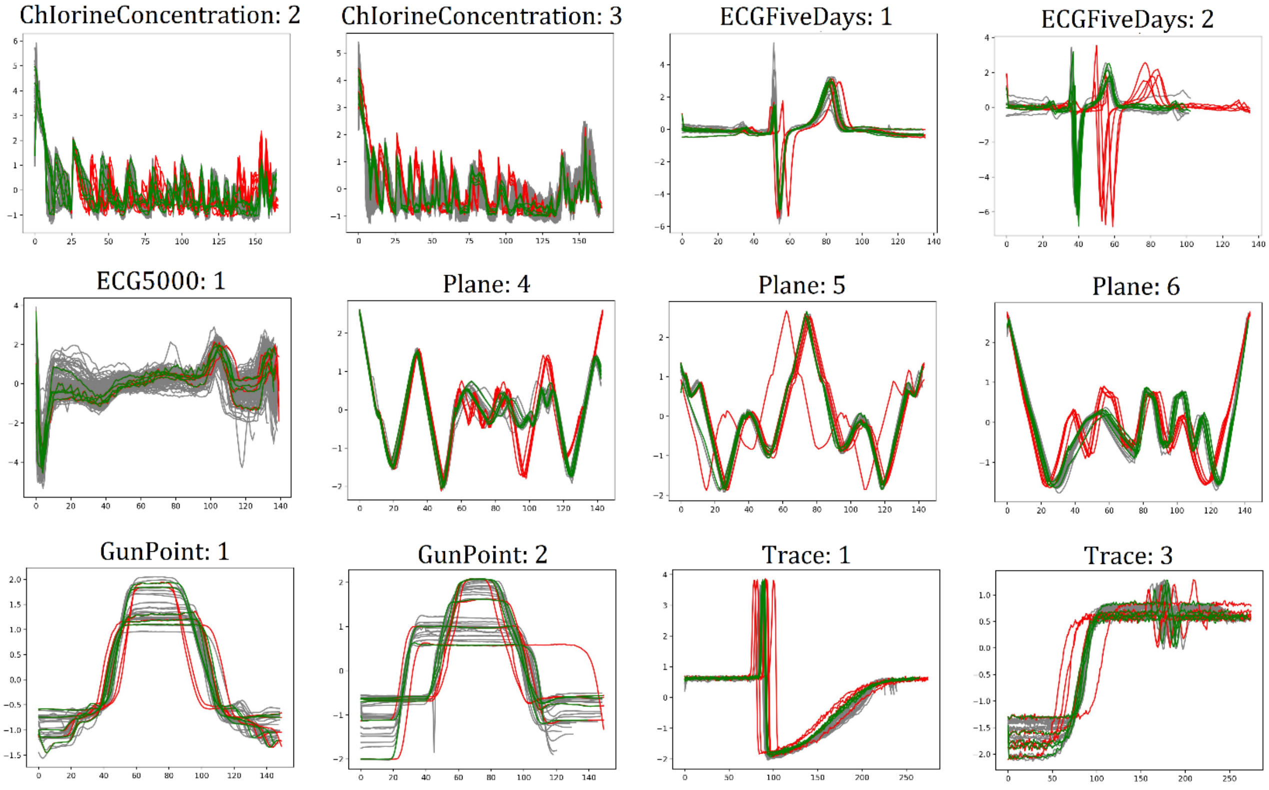


Fig. 4: Results of the MTSA experiment, with dataset names and labels displayed above each. In each plot, gray signals represent the warped training signals, while red signals indicate five randomly selected test signals requiring alignment. The green signals show the warped versions of the red signals, generated by our model.

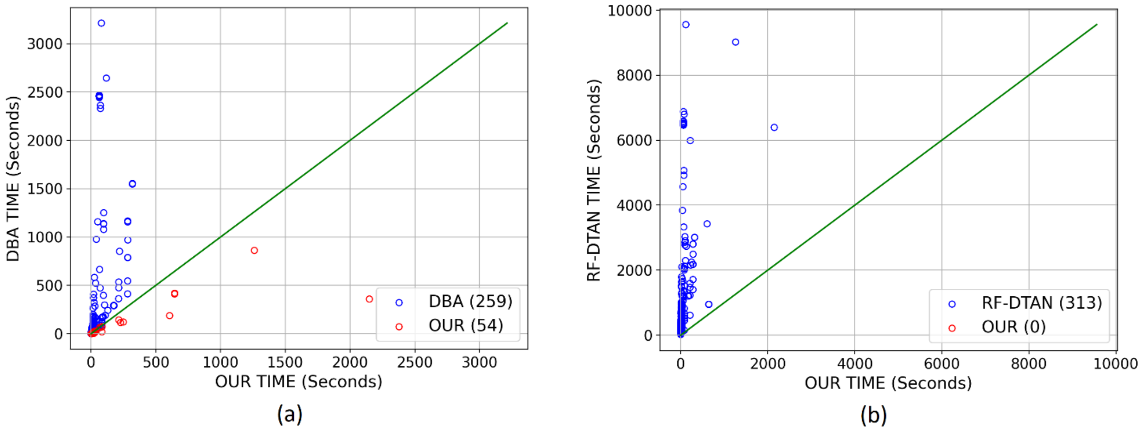


Fig. 5: Scatter plots to compare the timing of our method with: (a) DBA and (b) RF-DTAN. Each point represents a label from a dataset, with points above the $y = x$ line indicating a win (blue points) and those below showing a loss (red points) for our model.

Table 1: Timing comparison for the MTSA task between our method and the DBA and RF-DTAN approaches.

Dataset name	**Label**	**Number of Training signals**	**OUR time: Train (sec)**	**OUR Time: Test (sec)**	**OUR Time: Whole (sec)**	**DBA time: Whole (sec)**	**RF-DTAN time: Whole (sec)**
ChlorineConcentration	2	91	11.6	2.27	13.87	87.7	1009.2
ChlorineConcentration	3	262	102.4	2.24	104.6	259.9	2905.7
ECG5000	1	292	127.6	1.65	129.2	201.6	2732.9
ECGFiveDays	1	14	0.30	1.51	1.81	3.94	155.5
ECGFiveDays	2	9	0.13	1.55	1.68	2.04	100.0
GunPoint	1	24	0.81	1.89	2.7	11.4	253.5
GunPoint	2	26	0.95	2.02	2.97	12.9	274.7
Plane	4	16	0.38	1.73	2.11	5.41	155.2
Plane	5	13	0.25	1.71	1.96	3.97	126.1
Plane	6	18	0.46	1.77	2.23	6.56	174.6
Trace	1	26	0.96	6.15	7.11	42.3	484.5
Trace	3	22	0.70	6.15	6.85	32.4	409.9

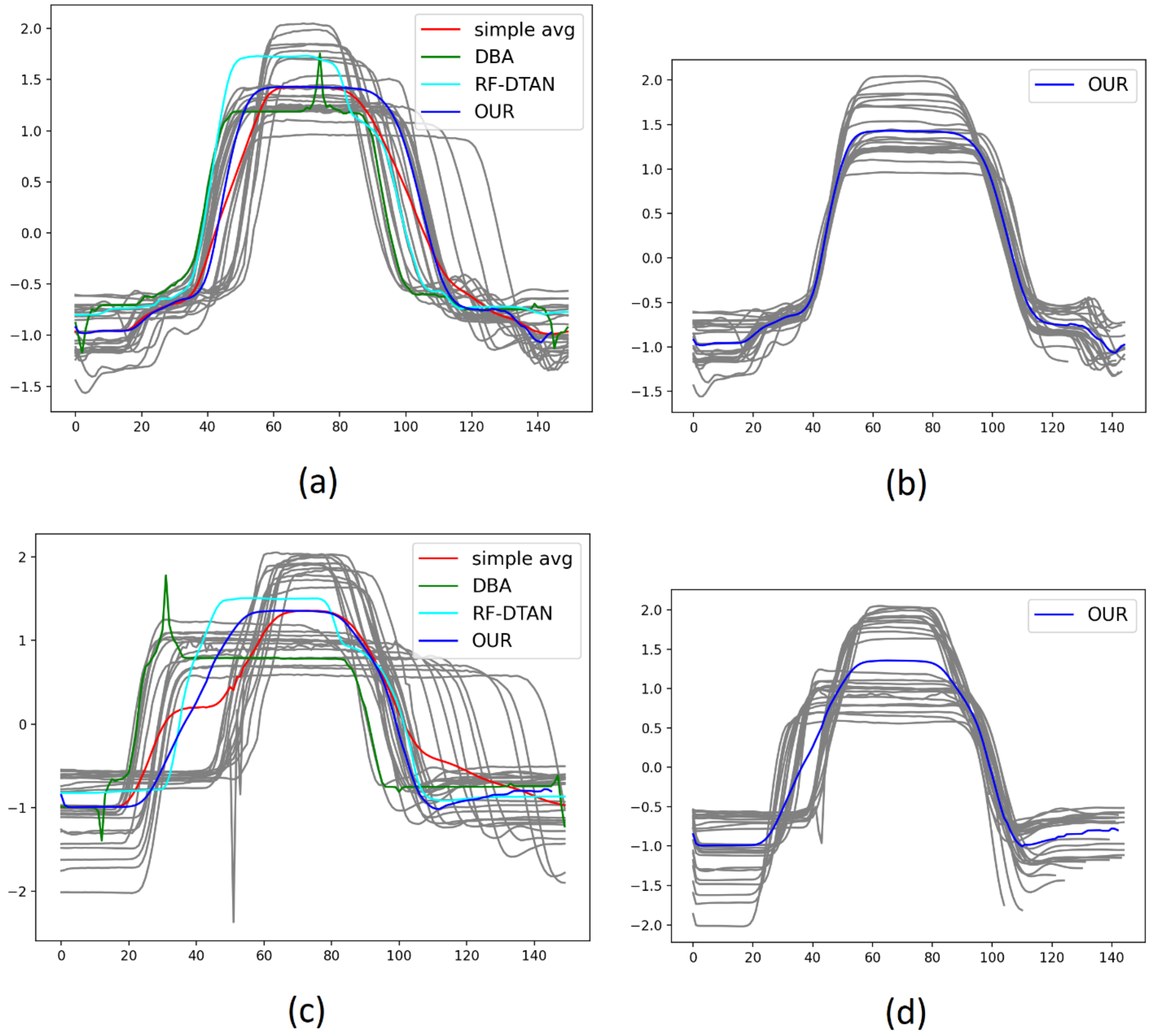


Fig. 6: Results on the GunPoint dataset. (a) label 1, gray: original time series, red: simple average, green: DBA, cyan: RF-DTAN, blue: OUR warped average. (b) label 1, gray: warped time series with OUR method, blue: OUR warped average. (c) label 2, similar to (a) for colors. (d) label 2, similar to (b) for colors.

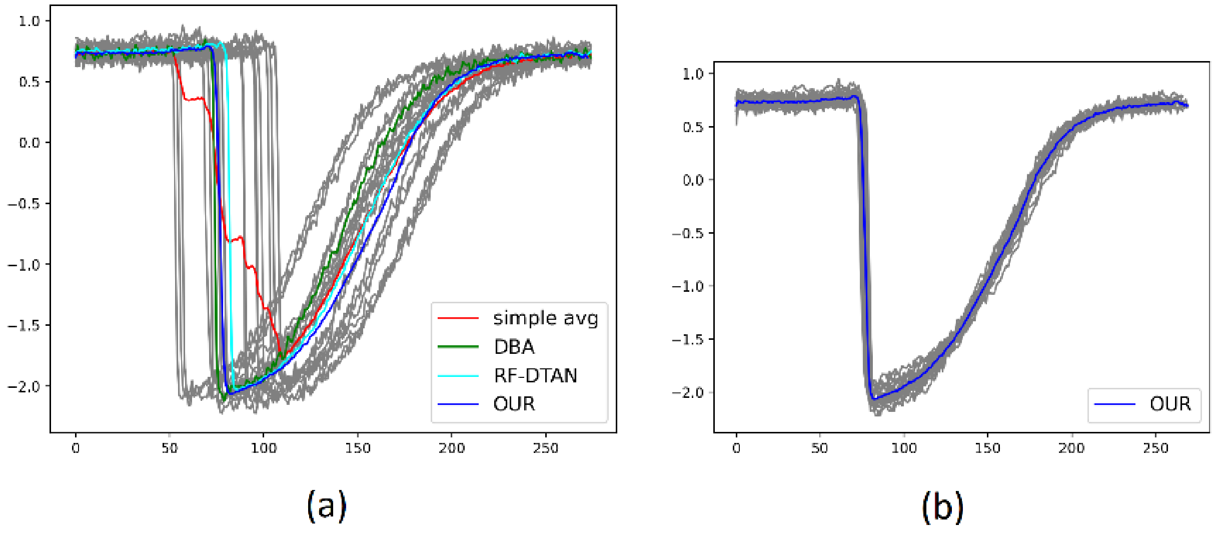


Fig. 7: Results on the Trace dataset, label 2. For details refer to Fig. 6 caption.

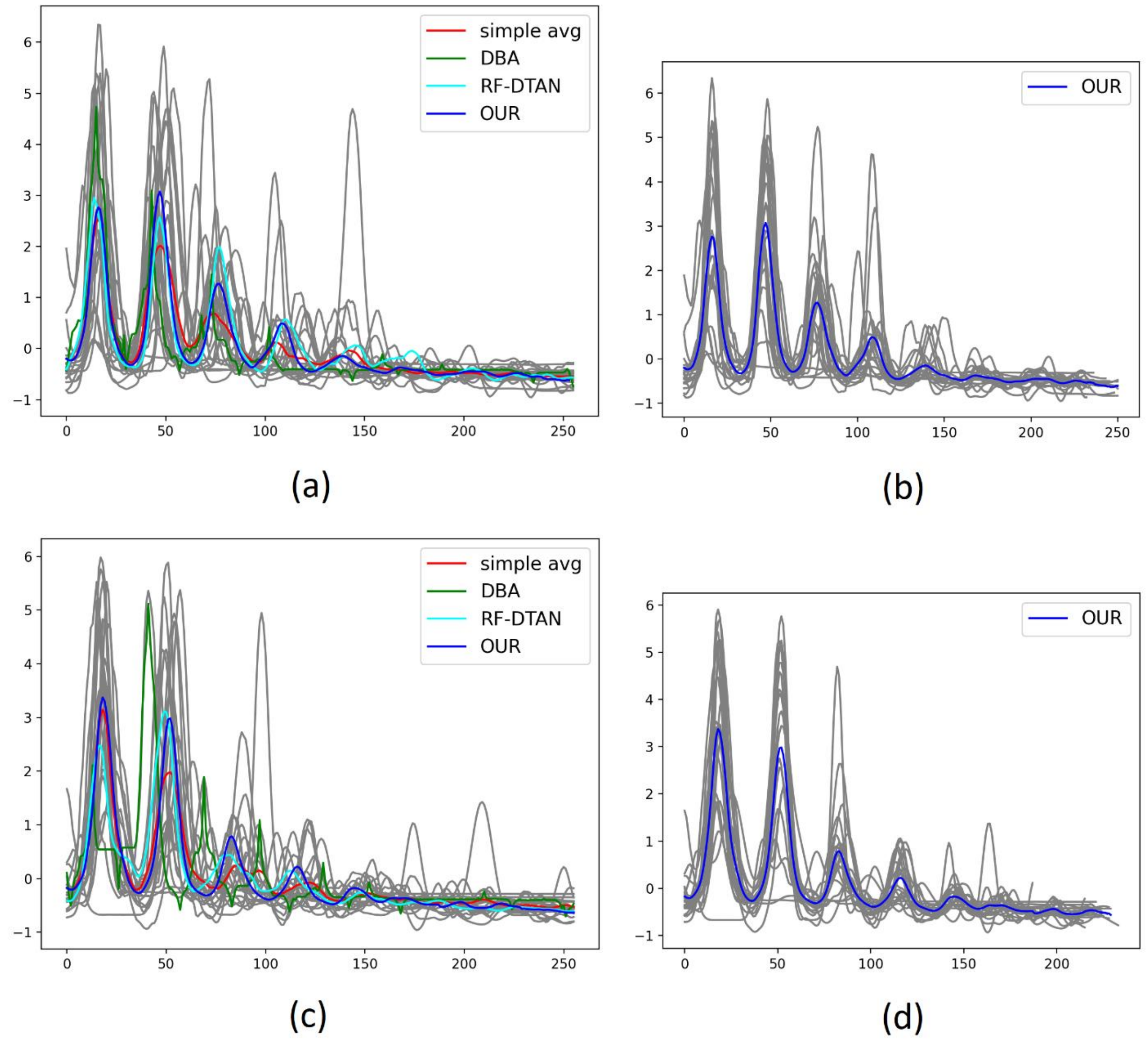


Fig. 8: Results on the InsectWingbeatSound dataset, (a), (b): label 2 and (c), (d): label 10. For details refer to Fig. 6 caption.

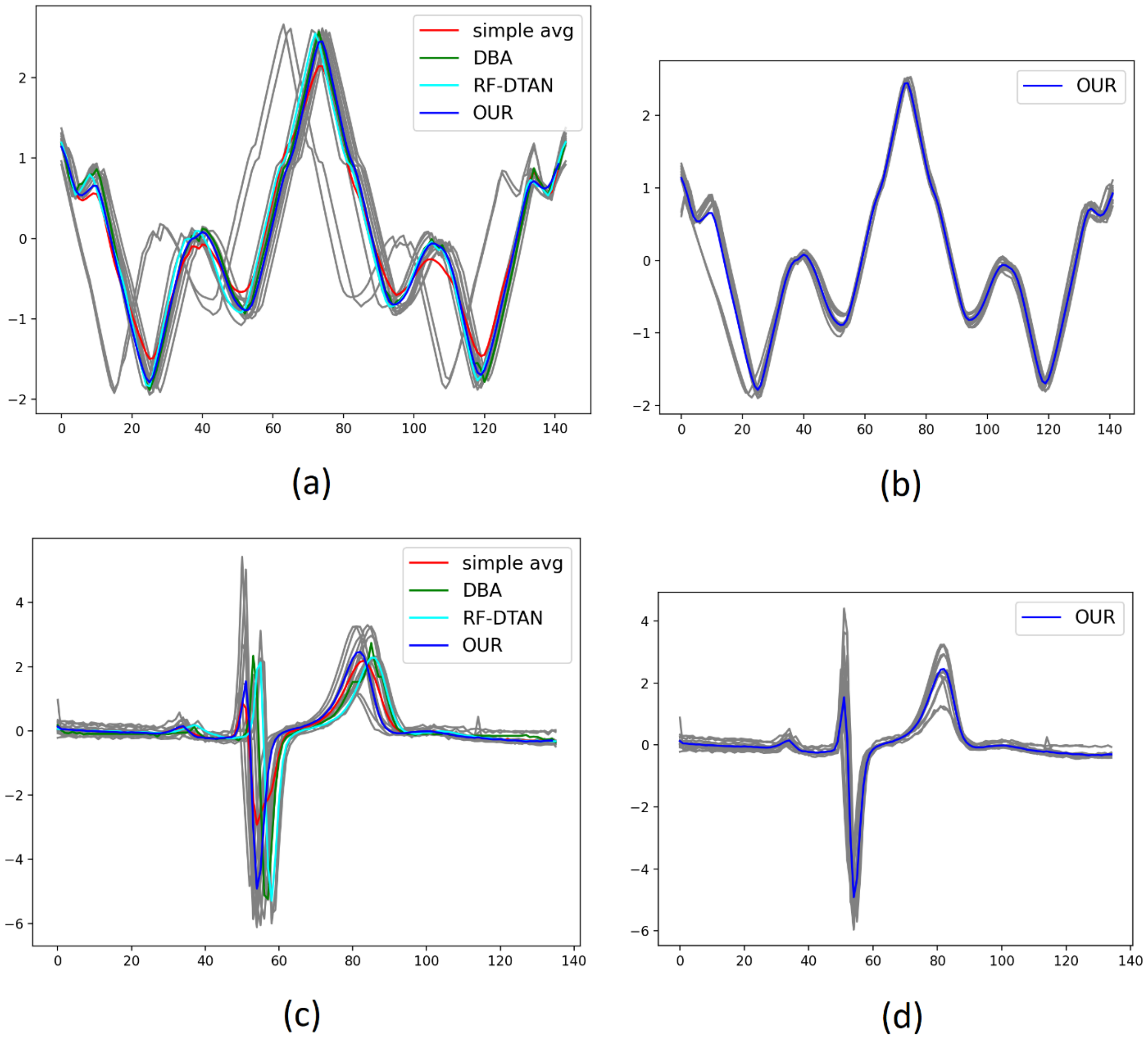


Fig. 9: (a), (b): Results on the Plane dataset, label 5. (c), (d): Results on the ECGFiveDays dataset, label 1. For details refer to Fig. 6 caption.

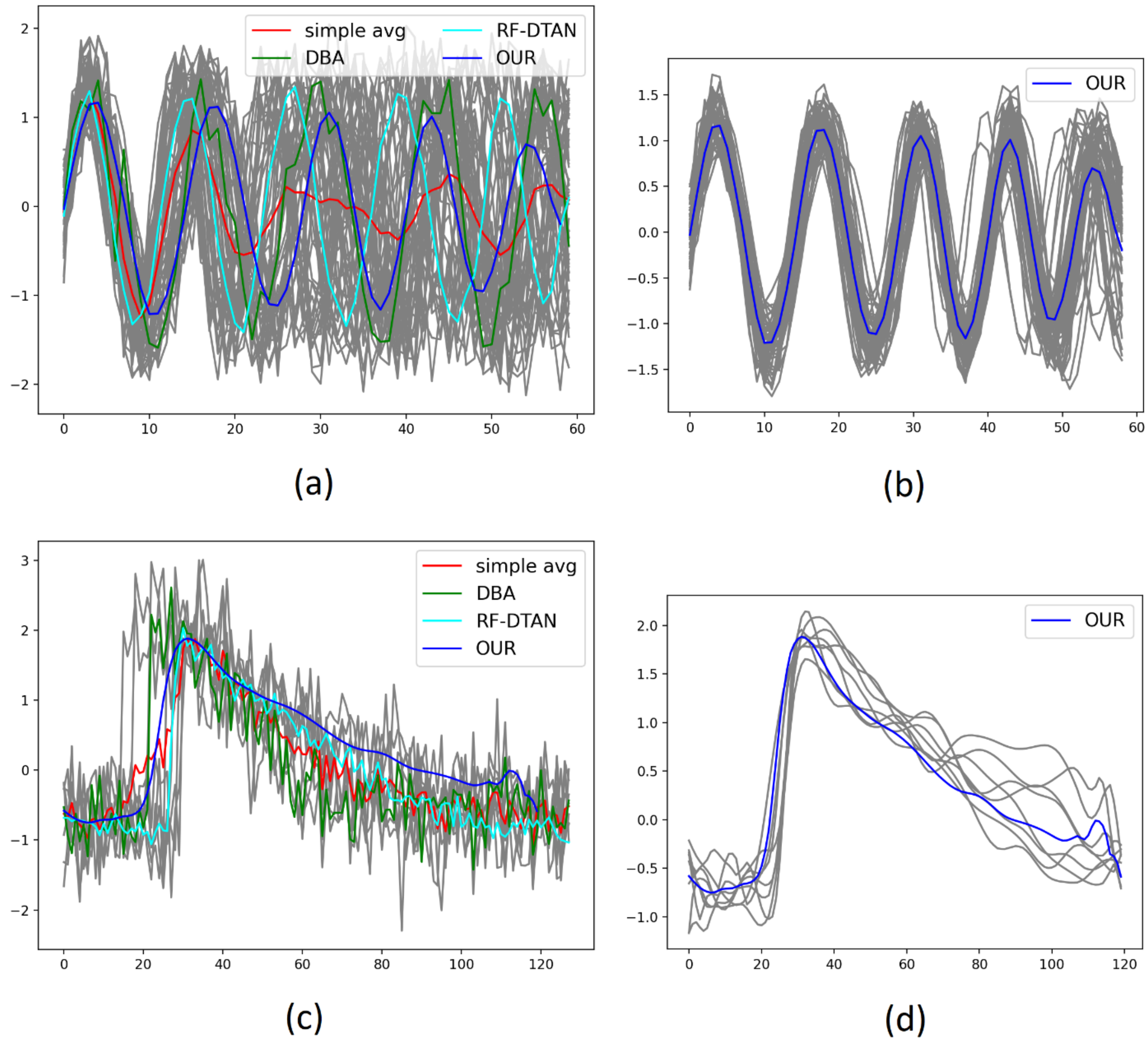


Fig. 10: (a), (b): Results on the SyntheticControl dataset, label 2. (c), (d): Results on the CBF dataset, label 3. For details refer to Fig. 6 caption.

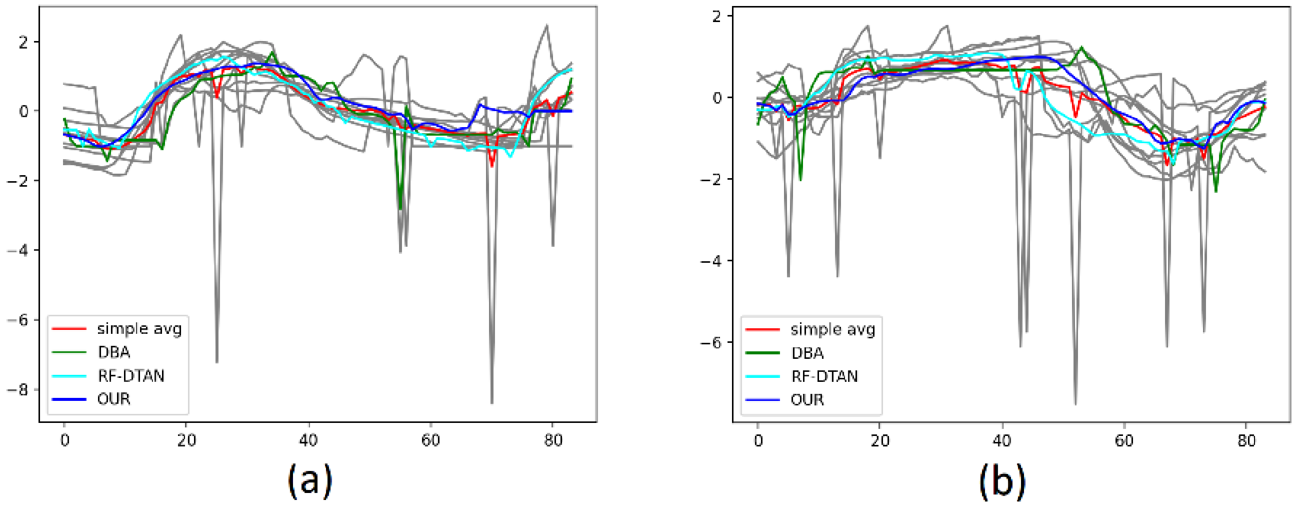


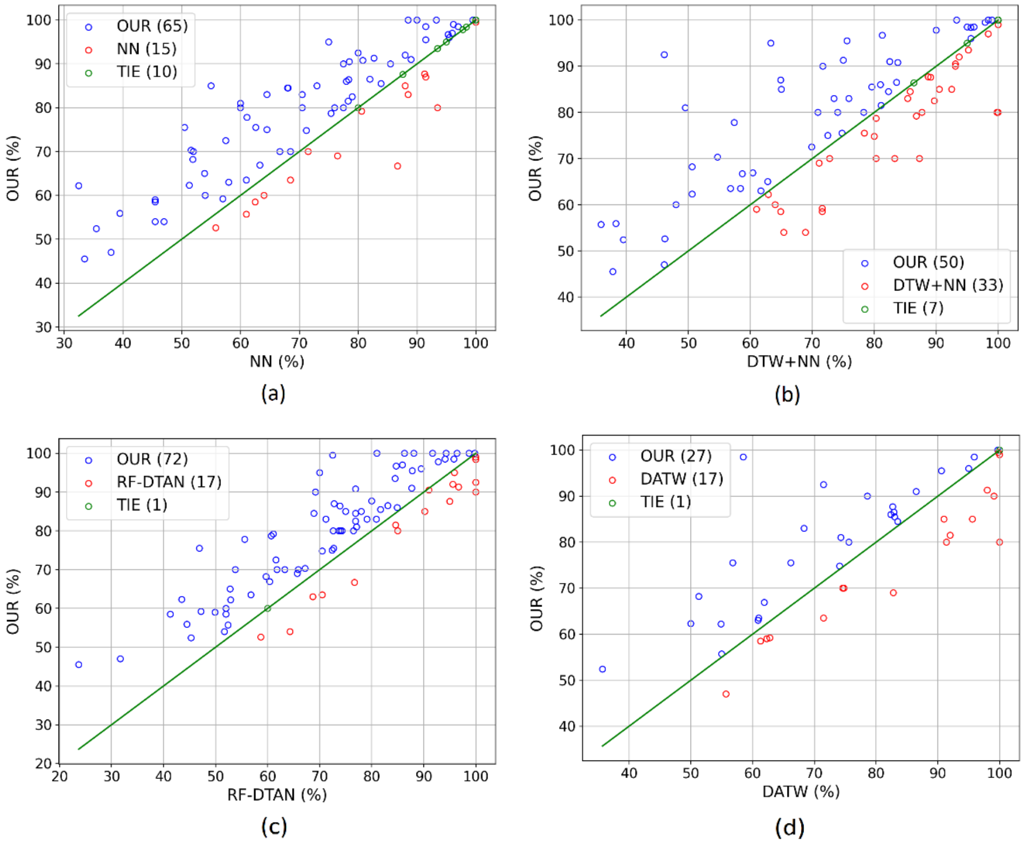
Fig. 11: Results on the MoteStrain dataset. gray: original time series, red: simple average, green: DBA signal, blue: warped average with our method. (a): label 1, (b): label 2.

Table 2: Classification accuracy comparison between our method and five baselines over 90 datasets of the UCR Archive.

Dataset name	Train	Test	Class	Length	Base	DTW+	DBA+	DATW	RF-	OUR	CS org . ->
					NN	NN	NN		DTAN		CS warp
ACSF1	100	100	10	1460	54	**64**	47	-----	52	60	0.326->0.022
Trace	100	100	4	275	76	**100**	86	**100**	74	80	0.392->0.042
CBF	30	900	3	128	85.5	71.7	92.2	99.1	**100**	90	0.745->0.141
TwoLeadECG	23	1139	2	82	78.5	**93.1**	87.1	-----	91	90.5	0.247->0.048
SmoothSubspace	150	150	3	15	95.3	81.3	82.7	-----	84.7	**96.7**	0.275->0.078
ECG200	100	100	2	96	88	**92.5**	83	91	78	85	0.275->0.085
SonyAIBORobotSurface2	27	953	2	65	**88.5**	85.4	76.4	-----	79.1	83	0.633->0.218
BME	30	150	3	128	82.7	75	75.3	**98**	96.7	91.3	0.268->0.120
Car	60	60	4	577	60	**99.8**	63.3	-----	85	80	0.148->0.068
GunPoint	50	150	2	150	**91.3**	88.7	76.7	82.7	80	87.7	0.366->0.167
Computers	250	250	2	720	57	**71.6**	56.8	62.8	47.2	59.2	0.955->0.452
InlineSkate	100	550	7	1882	33.5	37.8	31.6	-----	23.7	**45.5**	0.591->0.285
Plane	105	105	7	144	96.2	**100**	99	**100**	**100**	99	0.101->0.050
AllGestureWiimoteZ	300	700	10	Vary	47	**65.4**	53	-----	51.7	54	0.634->0.330
PhalangesOutlinesCorrect	1800	858	2	80	77.5	**93.1**	75.9	78.6	69.2	90	0.078->0.040
UMD	36	144	3	150	80.6	**86.8**	71.8	-----	61.1	79.2	0.299->0.161
GunPointAgeSpan	135	316	2	150	96	**98.4**	87.7	-----	85.9	97	0.046->0.025
ECGFiveDays	23	861	2	136	80	46.1	68.4	71.5	**100**	92.5	0.667->0.364
Fish	175	175	7	463	78.3	81.1	69.1	**92**	84.6	81.5	0.087->0.049
Chinatown	20	343	2	24	95	95	85.4	-----	**95.9**	95	0.371->0.215
InsectWingbeatSound	220	1980	11	256	**61**	35.9	40.9	55	52.4	55.7	0.657->0.403
FreezerRegularTrain	150	2850	2	301	79	**89.7**	77.1	-----	76.9	82.5	0.346->0.223
Yoga	300	3000	2	426	82	83.6	81.2	82.9	83.1	**86.5**	0.675->0.445
WormsTwoClass	181	77	2	900	61	58.4	54.5	61	56.8	**63.5**	0.918->0.609
ProximalPhalanxOutlineCorrect	600	291	2	80	77.5	78.3	74.6	**91.4**	74.3	80	0.034->0.023
SyntheticControl	300	300	6	60	88.5	99	92.3	99.7	98.7	**100**	0.655->0.446
MedicalImages	381	760	10	99	70.5	73.5	71.2	68.3	71.2	**83**	0.565->0.388
FreezerSmallTrain	28	2850	2	301	64.5	75.9	75.8	-----	80.9	**83**	0.290->0.200
Meat	60	60	3	448	93.3	93.3	90	-----	91.7	**100**	0.000->0.000
Herring	64	64	2	512	51.6	54.7	59.4	-----	67.2	**70.3**	0.090->0.064
Lightning7	70	73	7	319	57.5	69.9	68.5	-----	61.6	**72.5**	0.722->0.512
MiddlePhalanxOutlineCorrect	600	291	2	80	76.5	71.1	68.1	**82.8**	65.7	69	0.050->0.035
ECG5000	500	4500	5	140	91.5	75.6	84.5	90.6	87.8	**95.5**	0.374->0.270
FaceAll	560	1690	14	131	68	**85.8**	68.7	83.5	76.9	84.5	0.780->0.568
BirdChicken	20	20	2	512	55	65	65	-----	75	**85**	0.682->0.496
Wafer	1000	6164	2	152	**100**	97.9	92.5	99.8	72.5	99.5	0.518->0.382
Symbols	25	995	6	398	93.5	**95.2**	93.8	-----	84.5	93.5	0.212->0.158
Worms	181	77	5	900	45.5	61	45.5	**62.3**	49.9	59	0.899->0.672
ItalyPowerDemand	67	1029	2	24	97	95	92.7	95.9	94.1	**98.5**	0.312->0.240
MiddlePhalanxOutlineAgeGroup	400	154	3	80	51.9	50.6	57.1	51.3	59.7	**68.2**	0.030->0.023
MiddlePhalanxTW	399	154	6	80	51.3	50.6	48.7	50	43.5	**62.3**	0.018->0.014
ProximalPhalanxOutlineAgeGroup	400	205	3	80	78	81	81.5	82.4	84.9	**86**	0.021->0.017
DistalPhalanxTW	400	139	6	80	63.3	60.4	63.3	61.9	60.4	**66.9**	0.025->0.020
ArrowHead	36	175	3	251	**80**	70.9	67.1	-----	72.6	**80**	0.134->0.109
FordA	3601	1320	2	500	68.5	56.8	62.5	**71.5**	70.5	63.5	0.473->0.389
FordB	3636	810	2	500	58	61.7	61.1	60.9	**68.7**	63	0.481->0.396
DiatomSizeReduction	16	306	4	345	91.5	96.1	84.3	58.5	95.8	**98.5**	0.009->0.007
MoteStrain	20	1252	2	84	89	82.5	88.2	86.5	87.7	**91**	0.714->0.592
Strawberry	613	370	2	235	95.5	95.6	87.8	95	89.5	**96**	0.063->0.052
CinCECGTorso	40	1380	4	1639	**91.5**	64.9	63.2	-----	72.8	87	0.740->0.619
Wine	57	54	2	234	61.1	57.4	70.4	-----	55.6	**77.8**	0.002->0.002
Ham	109	105	2	431	60	49.5	71.4	74.3	77.1	**81**	0.440->0.370
SonyAIBORobotSurface1	20	601	2	70	64.5	72.5	71.7	-----	72.4	**75**	0.423->0.356
Haptics	155	308	5	1092	39.5	38.3	40.9	-----	44.5	**55.9**	0.430->0.365
ToeSegmentation2	36	130	2	343	80.8	83.8	80.8	-----	76.9	**90.8**	0.876->0.747
ProximalPhalanxTW	400	205	6	80	70.5	74.1	65.9	75.6	73.7	**80**	0.008->0.007
ChlorineConcentration	467	3840	3	166	62.5	**64.9**	53	61.3	52	58.5	0.311->0.271
AllGestureWiimoteY	300	700	10	Vary	45.5	**68.9**	57.1	-----	64.3	54	0.882->0.786
HouseTwenty	40	119	2	2000	68.1	82.3	83.2	-----	68.9	**84.5**	0.912->0.817
Lightning2	60	61	2	637	75.4	**80.3**	70.5	-----	60.7	78.7	0.554->0.498
AllGestureWiimoteX	300	700	10	Vary	45.5	**71.6**	54.3	-----	41.3	58.5	0.899->0.810
GunPointMaleVersusFemale	135	316	2	150	99.5	98.4	93.7	-----	94.3	**100**	0.052->0.047
ToeSegmentation1	40	228	2	277	68.5	**80.3**	64.9	-----	61.8	70	0.929->0.840
Beef	30	30	5	470	66.7	**87.3**	66.7	-----	63.3	70	0.233->0.214
FacesUCR	200	2050	14	131	73	90.5	82.5	**95.6**	90.2	85	0.753->0.693
Rock	20	50	4	2844	**64**	48	44	-----	60	60	0.849->0.790
PowerCons	180	180	2	144	**97.8**	90	95	-----	92.8	**97.8**	0.492->0.460
DistalPhalanxOutlineCorrect	600	276	2	80	71.5	72.8	72.5	**74.6**	65.9	70	0.136->0.128
OSULeaf	200	242	6	427	52	**83.3**	50.9	74.8	53.8	70	0.821->0.772
TwoPatterns	1000	4000	4	128	90	**100**	80.2	**100**	99.8	**100**	0.935->0.900
DistalPhalanxOutlineAgeGroup	400	139	3	80	62.6	74.8	69.8	66.2	72.7	**75.5**	0.109->0.105
GunPointOldVersusYoung	136	315	2	150	**100**	**100**	91.4	-----	88.1	**100**	0.048->0.047
BeetleFly	20	20	2	512	75	63.3	70	-----	70	**95**	0.946->0.914
ScreenType	375	375	3	720	35.5	39.5	44	35.7	45.3	**52.4**	0.941->0.923
LargeKitchenAppliances	375	375	3	720	50.5	**78.4**	49.1	56.8	46.9	75.5	0.984->0.967
DodgerLoopWeekend	20	138	2	288	98.4	95.6	97.7	-----	**100**	98.4	0.133->0.132
ShapeletSim	20	180	2	500	53.9	62.8	53.9	-----	52.8	**65**	0.998->0.990
SmallKitchenAppliances	375	375	3	720	32.5	62.9	**66.7**	54.9	52.9	62.2	0.996->0.989
Earthquakes	322	139	2	512	71.2	**80**	74	74.1	70.5	74.8	0.993->0.990
RefrigerationDevices	375	375	3	720	38	46.1	40.8	**55.7**	31.7	47	0.993->0.990
Fungi	18	186	18	201	83.9	79.6	**87.1**	83	81.7	85.5	0.000->0.000
FaceFour	24	88	4	350	78.4	**86.4**	81.8	-----	73.9	**86.4**	0.694->0.706
Mallat	55	2345	8	1024	88	93.7	94.8	-----	**95.6**	92	0.041->0.042
DodgerLoopGame	20	138	2	288	87.6	89.1	78.3	-----	**95**	87.6	0.156->0.176
InsectEPGRegularTrain	62	249	3	601	**100**	**100**	**100**	-----	86.3	**100**	0.025->0.029
DodgerLoopDay	78	80	7	288	55.8	46.2	49.4	-----	**58.7**	52.6	0.129->0.151
InsectEPGSmallTrain	17	249	3	601	**100**	**100**	**100**	-----	81	**100**	0.019->0.023
Coffee	28	28	2	286	**100**	**100**	96.4	-----	96.4	**100**	0.013->0.016
MelbournePedestrian	1194	2439	10	24	**93.5**	87.7	71.4	-----	76.5	80	0.122->0.322
OliveOil	30	30	4	570	**86.7**	58.7	**86.7**	-----	76.7	66.7	0.000->0.002
Average	-----	-----	-----	-----	73.6	76.6	72.2	76.7	73.5	**79.7**	0.425->0.330
MPCE	-----	-----	-----	-----	0.0832	0.0760	0.0881	0.0773	0.0861	**0.0627**	--------------

Table 3: Results of the Wilcoxon signed-rank test to assess the statistical significance of our approach over five baselines.

Baseline	p-value	95% Confidence Interval of accuracy improvement (%)
NN	1.45×10^{-8}	[4.37, 7.97]
DTW+NN	0.018	[0.86, 5.52]
DBA+NN	6.33×10^{-13}	[5.88, 9.03]
RF-DTAN	2×10^{-9}	[4.82, 8.21]
DATW	0.275	[-0.84, 5.29]



(a)

(b)

(c)

(d)

Fig. 12: Scatter plots comparing our method with (a) NN, (b) DTW+NN, (c) RF-DTAN and (d) DATW. Each point represents a dataset, with points above the $y = x$ line indicating a win (blue points) and those below showing a loss (red points) for our model.

Table 4: RESNET Test Loss Average and Variance percentage improvements over epochs (after epoch 300) and Accuracy comparison for 30 datasets when a warping stage with our approach is added.

Dataset name	**% Loss Avg. Improvement**	**% Loss Var. Improvement**	**% Acc. Without Pre-warping**	**% Acc. With Pre-warping**
Birdchicken	50.4	92.8	85	**95**
BME	81.5	62	98.7	**100**
CBF	62.8	21.5	**99.4**	**99.4**
Coffee	40.4	85.7	**100**	**100**
DistalPhalanxTW	-23.6	35.6	68.3	**71.2**
DodgerLoopGame	37.4	97.5	48.8	**51.2**
Earthquakes	3.7	-24.1	69.1	**75.5**
ECG5000	8.9	-53.4	93.3	**93.6**
FaceFour	14.2	86.1	**95.4**	94.3
FreezerRegularTrain	72.9	100	**99.8**	98.7
GunPoint	91.9	100	98.7	**99.3**
GunPointOldVersusYoung	99	100	97.8	**100**
Herring	34.1	79.9	60.9	**65.6**
LargeKitchenAppliances	-40.8	21.7	81.1	**90.4**
Lightning2	20.3	97.2	77	**83.6**
Mallat	5.2	-20.6	91.2	**97.4**
MoteStrain	40.2	70.4	91.4	**93.7**
PowerCons	42.1	74.3	86.1	**90**
ProximalPhalanxOutlineAgeGroup	-3.1	57.3	82.9	**88.3**
ProximalPhalanxOutlineCorrect	17.7	-7.8	91.4	**93.1**
RefrigerationDevices	-8.8	52	51.7	**53.1**
SonyAIBORobotSurface1	-30.1	23.2	93.3	**94**
Symbols	13	-102.4	91	**95.5**
SyntheticControl	69.3	100	**99.3**	98.7
ToeSegmentation1	42	95.9	96.9	**98.7**
Trace	99.7	100	**100**	**100**
TwoLeadECG	-3.9	12.4	**100**	**100**
TwoPatterns	88.3	100	95.9	**99.7**
UMD	10	72.3	98.6	**99.3**
Wafer	58.5	99.7	**99.8**	99.7
Average	**33.1**	**54.3**	88.1	**90.6**
MPCE	—	—	0.0374	**0.0289**

Alireza Nourbakhsh received B.Sc. and M.Sc. degrees in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2017 and 2019, respectively. He is currently pursuing a Ph.D. degree in electrical engineering with Digital Electronics Group, Sharif University of Technology, Tehran, Iran.

Hoda Mohammadzade is an associate professor at Sharif University of Technology, Department of Electrical Engineering. She received her BSc degree from Amirkabir University of Technology (Tehran Polytechnic), Iran, in 2004, the MSc degree from the University of Calgary, Canada, in 2007, and the PhD degree from the University of Toronto, Canada, in 2012, all in electrical engineering. Her research interests include machine learning, computer vision, biomedical signal processing, and biometrics. She has published several papers in international journals and conferences.